In the Claims

A1

1(Original). A search engine system, comprising:

an associative memory;

a first search engine having a first data input and a connection to the associative memory; and

a second search engine having a second data input and a connection to the associative memory.

2(Original). The search engine system of claim 1, further including a pre-parser having an input connected to the first data input and an output connected to an input of the first search engine.

3(Original). The search engine system of claim 1, further including a hit output queue connected to the first search engine.

4(Original). The search engine system of claim 3, further including a proximity search engine connected to an output of the first search engine.

5(Original). The search engine system of claim 4, further including a key list connected to the proximity search engine.

6(Original). The search engine system of claim 5, further including a proximity hit queue connected to the proximity search engine.

7(Original). The search engine system of claim 1, wherein the first search engine includes a transform generator.

8(Original). The search engine system of claim 7, wherein the transform generator converts an input data into an address and a confirmer.

A1

9(Original). The search engine system of claim 5, wherein the key list contains at least two text strings and a distance between the at least two text strings.

10(Original). The search engine system of claim 1, further including a packet input queue connected to the associative memory.

11(Original). A method of operating a search engine system, comprising the steps of:

a) forming a packet of data;

b) when the packet of data contains a start flag, starting a sliding window search on the packet of data;

c) when a match is found, determining a location of the match.

12(Original). The method of claim 11, wherein step (a) further includes the step of:

a1) parsing a raw data to find a predetermined set of characters;

a2) when the predetermined set of characters is found, replacing the predetermined set of characters with a replacement set characters.

13(Original). The method of claim 12, wherein step (a1) further includes the steps of:

i) defining the predetermined set of characters to be any combination of white space characters;

ii) defining the replacement set of characters as a space character.

A1

14(Original). The method of claim 12, wherein step (a1) further includes the steps of:

i) defining the predetermined set of characters to be all capital letter;

ii) defining the replacement set of characters as a corresponding lower case letter.

15(Original). The method of claim 11, further including the steps of:

d) determining if the match is contained in a proximity key list;

e) when the match is contained in a proximity key list, determining if the match is a primary index;

f) when the match is a primary index, storing the match in the proximity hit queue.

16(Original). The method of claim 15, further including the step of:

f) when the match is a next index, searching the proximity hit queue for an associated primary index.

17(Original). The method of claim 16, further including the steps of:

g) determining if a first entry is the associated primary index;

h) when the first entry is the associated primary index, determining a distance between the next index and the primary index;

i) when the distance between the next index and the primary index is less than a proximity offset storing a proximity hit in the final proximity hit queue.

A1

18(Original). The method of claim 16, further including the steps of:

j) when the distance between the next index and the primary index is not less than the proximity offset, flushing the primary index from the proximity hit queue.

19(Original). The method of claim 11, wherein step (a) further includes the steps of:

a1) receiving an input data stream;

a2) removing an overhead data to form a raw data stream;

a3) determining a start of a message;

a4) forming a search packet containing a start flag and a portion of the raw data;

a5) forming a plurality of search packets containing only the raw data;

a6) determining an end of the message;

a7) when an end of message is found, forming a final search packet containing an end flag.

20(Original). A search engine system comprising:

an associative match memory;

a sliding search engine connected to the associative match memory; and

a proximity search engine connected to an output of the sliding search engine.

21(Original). The search engine system of claim 20, further including a plurality of sliding search engines that are each connected to a separate data stream.

22(Original). The search engine system of claim 20, further including a plurality of proximity engines are connected the sliding search engine.

A1

23(Original). The search engine system of claim 20, further including a pre-parser connected to one of the sliding search engine.

24(Original). The search engine system of claim 23, wherein each of the plurality of pre-parsers contains a mapping table.

25(Original). The search engine system of claim 24, wherein an entry in the mapping table contains a characters to be replaced location and a replacement characters location.

26(Original). The search engine system of claim 20, further including a plurality of proximity search engines connected to the sliding search engine.

27(Original). The search engine system of claim 26, further including a key list memory connected to the proximity search engine.

28(Original). The search engine system of claim 27, wherein the key list contains a plurality of locations, at least one of the plurality of locations contains a primary index, a next index and a proximity offset.

29(Original). The search engine system of claim 27, wherein the proximity search engine contains a proximity hit list.

30(Original). The search engine system of claim 21, further including a data input processor.